



Droit et Conduite de Projet

Initiation à Python

Résponsables : M. Maxime MARIA & M. Emmanuel CONCHON

Etudiants : Kamelia BENSMAIL & Badr OUARAFANA & Vincent
JOULAIN

Année universitaire : 2022 - 2023

Contents

1	Projet d'initiation a Python	3
2	Première rencontre avec la BFM de l'Aurence	3
3	Préparation des Ateliers	3
3.1	1er atelier - Tic Tac Toe	3
3.2	2e atelier - Brute Force	5
4	Deuxième rencontre avec la BFM de l'Aurence	6
5	Déroulement des séances	6
6	Retour sur expérience	7

Contents

1 Projet d'initiation a Python

Nous avons décidé de sélectionner le projet proposé par la BFM de l'Aurence, une activité d'initiation python à l'intention d'un public de tout âge. Ces ateliers sont dans la continuité de ceux réalisés les années précédentes et permettent ainsi à la bibliothèque de proposer des activités variées à son public.

Nous nous sommes dirigées vers ce projet car il nous semblait être un moyen efficace de partager nos passions et nos connaissances à un public néophyte, en plus de développer nos capacités de pédagogie.

2 Première rencontre avec la BFM de l'Aurence

La première rencontre fut en mi-octobre avec Mme Anne TRICARD et M Olivier DELAUNAY qui nous ont agréablement accueilli et avec lesquels nous avons pu discuter :

- des disponibilités de chacun (choix de deux créneaux, 14 et 22 février)
- de la gestion de la publicité (création des affiches et publications)
- du public visé (nous nous sommes accordés sur un âge de 8 ans et plus)
- de la capacité d'accueil de la salle (8 postes pour 12 personnes maximum)
- de la durée des séances (deux heures)

Les choix effectués n'ont pas été forcément évidents, par exemple, définir un âge minimum de 8 ans renvoie l'impression que les plus âgés ne sont pas concernés par cette activité, or notre objectif était de toucher un public le plus large possible (différents niveaux de difficulté des activités proposées). Le choix des dates était lui aussi important, nous avons convenus de placer les créneaux lors des vacances scolaires de février afin de ne pas se retrouver à court d'intéressés.

Pour la publicité, ils nous ont précisé que l'atelier était consacré au public régulier de la bibliothèque et qu'il n'y avait donc aucunement besoin d'en faire la diffusion à l'extérieur de celle-ci (le public avait été réceptif les années précédentes).

3 Préparation des Ateliers

Une fois les variables extérieures définies, il nous a fallu passer au choix et développement d'un sujet. Nous avons pris le risque de créer deux projets différents afin de s'adapter au "tout public" que nous recherchions, proposer plusieurs niveaux de difficulté dans l'objectif de permettre à chacun de profiter au mieux de cette séance. Le risque était évidemment de scinder le groupe en plusieurs et de ne pas encourager à l'entraide parmi les participants (ce que nous ne souhaitons pas).

Nous avons dans un premier temps commencé par créer une fiche python débutant (disponible ici) que nous avons imprimé et distribué en début de séance afin de permettre aux participants de suivre une courte présentation en direct (sur laquelle nous revenons au fur et à mesure de l'avancement général) :

3.1 1er atelier - Tic Tac Toe

Le Tic Tac Toe était le projet destiné à un public plus jeune/débutant, l'objectif de ce jeu était la compréhension et l'utilisation des diverses fonctionnalités de Python telles que les variables, les booléens, les branchements, etc.

Le programme a été préparé à l'avance, et expliqué afin d'enseigner comment afficher du texte lié aux joueurs, à leur position et leur statut (gagnant ou perdant), d'ajouter certaines fonctionnalités (par exemple, la sélection de cases dans le jeu, la demande d'une ligne et d'une colonne) et de procéder à des vérifications (par exemple, si le joueur a gagné ou s'il a choisi une valeur incorrecte).

Au commencement, le jeu affiche le tableau suivant :

```
      [ 1 | 2 | 3 ]
[1] [ . | . | . ]
[2] [ . | . | . ]
[3] [ . | . | . ]
```

Chaque joueur choisit une ligne et colonne tour par tour :

```
      [ 1 | 2 | 3 ]
[1] [ x | . | . ]
[2] [ . | 0 | . ]
[3] [ . | . | . ]
```

Dans un premier temps il a fallu compléter la fonction `choisirCase()` :

- Utiliser la fonction `input()` pour demander la ligne et la colonne.
- Utiliser les branchements conditionnels afin de vérifier si une ligne/colonne est entre `[1, 3]` et si une case est vide.
- Manipuler les variables et affecter la valeur choisie par l'utilisateur dans le tableau (La syntaxe d'un tableau a été fournie).

```
def choisirCase(tableau, joueur , caractere):
    input()
    #Demandez à l'utilisateur de choisir une ligne et une colonne.
    #==>

    #Verifiez que la ligne et la colonne est bien un chiffre entre 1 et 3.
    #==>

    #Verifiez que la case n'est pas remplie. pour accéder à une case d'un tableau vous utiliserez la syntaxe suivante
    # variable = tableau[ligne-1][colonne-1]
    #==>

    #Affecter le caractère passé en paramètre au tableau[ligne-1][colonne-1]
    #==>
```

Pour celles et ceux qui ont réussi la première partie, la difficulté a été légèrement augmentée sur la suivante :

Le nouvel objectif est de terminer le jeu, c'est-à-dire afficher le joueur gagnant, changer le tour ainsi que le caractère choisi pour chacun des joueur.

```

def jouer(tableau):
    #le joueur et le caractère sont choisis aléatoirement sous forme d'un entier (1 ou 2) et un char ('x' ou 'o')
    joueur = random.randint(1,2)
    caractere = random.choice(["x" , "o"])
    while True :
        #Affichez le joueur en utilisant un print.
        #==>

        print()
        afficherTableau(tableau)
        choisirCase(tableau , joueur , caractere)
        gagnant = verifierLignes(tableau)
        if gagnant :
            print("joueur " , joueur , "à gagné, Bravo !")
            afficherTableau(tableau)
            return
        if verifierEgalite(tableau) :
            print("Egalité :\\")
            afficherTableau(tableau)
            return

    #Changez le tours des joueurs
    #==>

    #Changez caractère
    #==>

```

3.2 2e atelier - Brute Force

Le brute force était le projet destiné aux débutants les plus confirmés, composé de trois niveaux de difficulté croissante, l'objectif était de trouver le mot de passe à l'aide d'un dictionnaire ou d'une recherche exhaustive :

- Premier niveau, boucler sur une liste :

```

def bruteforce1(dictionnaire, motDePasse):
    for el in dictionnaire:
        testerMotdePasse(el, motDePasse)

```

- Deuxième niveau, générer toutes les possibilités sur un alphabet court :

```

def bruteforce2(chiffres, motDePasse):
    for el1 in chiffres:
        testerMotdePasse(el1, motDePasse)
        for el2 in chiffres:
            testerMotdePasse(el1 + el2, motDePasse)
            for el3 in chiffres:
                testerMotdePasse(el1 + el2 + el3, motDePasse)

```

- Troisième niveau, généraliser le raisonnement à l'aide d'une fonction récursive :

```

def bruteforce3(debut, alphabet, longueur, motDePasse):
    if len(debut) == longueur:
        return
    for el in alphabet:
        essai = debut + el
        testerMotdePasse(essai, motDePasse)
        bruteforce3(essai, alphabet, longueur, motDePasse)

```

Le code fourni aux participants était prêt à être exécuté, la seule chose à faire était de construire les fonctions bruteforce 1, 2 et 3 dans l'ordre afin de trouver tous les mots de passes (pour des raisons de simplicité ceux-ci étaient simplement encodés en UTF-8)

```

niveau1 = b"\x74\x6f\x6d\x61\x74\x65"
dictionnaire = ["cerise", "poire", "tomate", "ecureuil", "minecraft", "vroomvroom", "supermotdepasse",
                "impossibleletutrouves"]

```

```

niveau2 = b"\x31\x31\x33"
chiffres = ["1", "2", "3"]

niveau3 = b"\x79\x6f\x75\x70\x69"
alphabet = ["p", "y", "u", "i", "o"]

def bruteforce1(dictionnaire, motDePasse):
    # boucler sur le dictionnaire et tester toutes les possibilités
    return

def bruteforce2(chiffres, motDePasse):
    # comment faire pour afficher toutes les possibilités avec 1, 2 et 3 ?
    # (peut être utiliser plusieurs boucles)
    return

def bruteforce3(debut, alphabet, longueur, motDePasse):
    # généraliser le raisonnement (plus difficile)
    return

def testerMotdePasse(essai, motDePasse):
    if (essai == motDePasse.decode()):
        print("mot de passe trouvé : " + essai)
    return

bruteforce1(dictionnaire, niveau1)
bruteforce2(chiffres, niveau2)
bruteforce3("", alphabet, 5, niveau3)

```

4 Deuxième rencontre avec la BFM de l'Aurence

Le second rendez-vous que nous avons eu avec les membres de la BFM de l'Aurence, encadrants de notre projet, s'est déroulé début février afin de vérifier tous les points techniques à mettre en place :

- Essai des différents interpréteurs en ligne
- Essai de nos programmes
- Vérification du matériel (postes fournis, vidéo-projecteur, connectique ...)

Mais aussi de nous permettre de discuter du déroulement des ateliers avec les encadrants ainsi que de mettre au point les derniers préparatifs.

5 Déroulement des séances

Le public s'est trouvé être sensiblement différent lors de nos deux créneaux, nous avons accueilli la première séance des personnes d'âges variés (allant de 8 à 40 ans), les plus jeunes étaient accompagnés de leurs parents et la plupart d'entre eux ne se connaissaient pas. Il nous a été simple d'établir un climat calme tout au long de la séance.

Lors de la seconde séance ce fut un centre de loisir qui a été invité, les personnes présentes étaient d'âges similaires (entre 10 et 14 ans) et très familières les unes avec les autres, les encadrants du centre ne sont pas restés pendant le créneau ce qui ne nous a pas rendu la tâche plus simple.

Les deux séances ont suivi un déroulement similaire. Les activités, les discussions et les objectifs étaient identiques. Toutefois tous les participants ne sont pas arrivés jusqu'au bout au cours de la deuxième séance.

Suite à la courte présentation d'introduction, nous avons accordé un peu de temps aux participants afin de leur permettre de lire le code, essayer de comprendre ce qui a été expliqué et tester les premiers exemples. Durant

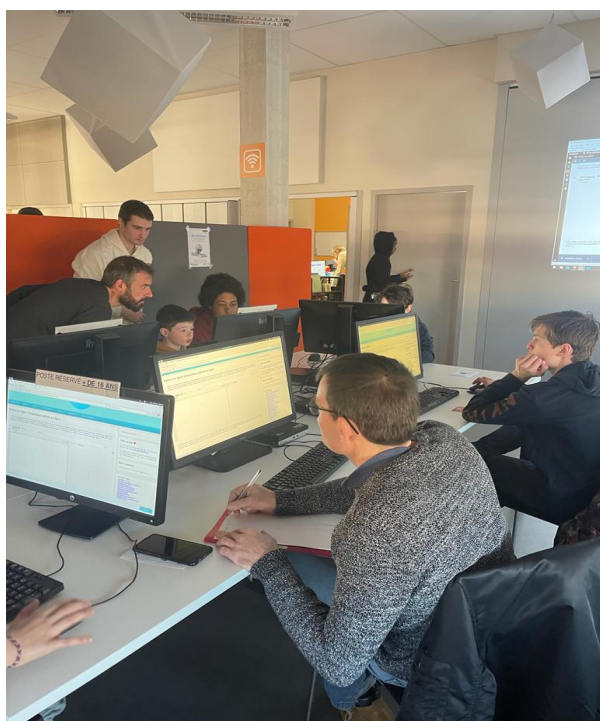
la séance, chacun d'entre nous à eu un échange individuel avec les participants en difficulté afin de discuter des problèmes rencontrés ainsi que de répondre aux questions.

Beaucoup des membres inscrits on rencontrés des difficultés pour comprendre le fonctionnement du programme choisi, mais une fois les exemples montrés (Demander une saisie, l'enregistrer ...), la majorité se sont montrés attentifs aux explications et les problèmes rapidement réglés.

Un suivi a été effectué au fil de l'avancement de chaque membre, dans le but de fournir une aide appropriée à ceux qui ont rencontré des difficultés. Ainsi, une attention particulière a été accordée à chaque personne tout au long du processus.

En fin de séance, la plupart des membres on pu comprendre l'idée derrière les programmes proposés, et ont pu finir la totalité des exercices demandés dont certains nous ont demandé comment approfondir de leur coté les prémices de connaissances vues dans ce cours.

Il est possible de retrouver les ateliers, fiches et corrections dans ce répertoire Github.



6 Retour sur expérience

Nous sommes très fiers d'avoir pu participer à ce projet, que ce soit grâce au remerciements des membres de la BFM pour leur avoir proposé une activité un peu différente à laquelle certains d'entre eux ont même participé, mais aussi au plaisir que nous avons eu à partager nos intérêts avec certains élèves motivés (qui se sont réinscrit pour le deuxième créneau) avec lesquels nous avons pu discuter à la fin des séances.

Toutefois nous nous sommes rendus compte de la difficulté d'encadrer un jeune groupe sur une durée de deux heures, la perte d'attention inhérente aux enfants de cet âge (notamment en groupe) et leur perte d'intérêt pour le sujet à la présentation d'une difficulté nous ont fait reconsidérer l'âge minimum que nous avons défini.

Nous avons travaillé en commun sur la majorité du projet (présence aux rendez-vous, développement des ateliers, entraide lors des séances ...), ce fut dans l'ensemble une bonne expérience pour nous trois.